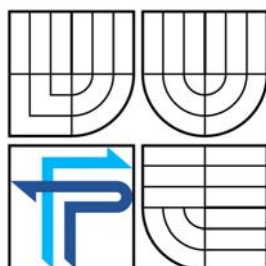


**VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ**  
BRNO UNIVERSITY OF TECHNOLOGY



**FAKULTA PODNIKATELSKÁ**  
**ÚSTAV INFORMATIKY**

FACULTY OF BUSINESS AND MANAGEMENT  
INSTITUT OF INFORMATICS

# **NÁVRH FIREMNÍHO DATABÁZOVÉHO SYSTÉMU ZBOŽÍ**

DATABASE DESIGN OF FIRM GOODS

**BAKALÁŘSKÁ PRÁCE**  
BACHELOR'S THESIS

**AUTOR PRÁCE**  
AUTHOR

**RADEK ZATOČIL**

**VEDOUCÍ PRÁCE**  
SUPERVISOR

**Ing. JIŘÍ KŘÍŽ, Ph.D.**

BRNO 2007

# **ZADÁNÍ BAKALÁŘSKÉ PRÁCE**

**Radek Zatočil**

---

6209R021 - Manažerská informatika

Ředitel ústavu v souladu se zákonem č. 111/1998 o vysokých školách, Studijním a zkušebním řádem VUT v Brně a Směrnicí děkana pro realizaci bakalářských a magisterských studijních programů Vám zadává bakalářskou práci s názvem:

**Návrh firemního databázového systému zboží**

**Database design of firm goods**

**Pokyny pro vypracování:**

Úvod

Vymezení problému a cíle práce

Teoretická východiska práce

Analýza problému a současné situace

Vlastní návrhy řešení, přínos (efektivnost) návrhů řešení

Závěr

Seznam použité literatury

Přílohy

---

Podle § 60 zákona č. 121/2000 Sb. (autorský zákon) v platném znění, je tato práce "Školním dílem". Využití této práce se řídí právním režimem autorského zákona. Citace povoluje Fakulta podnikatelská Vysokého učení technického v Brně. Podmínkou externího využití této práce je uzavření "Licenční smlouvy" dle autorského zákona.

# LICENČNÍ SMLOUVA

## POSKYTOVANÁ K VÝKONU PRÁVA UŽÍT ŠKOLNÍ DÍLO

uzavřená mezi smluvními stranami:

### 1. Pan/paní

Jméno a příjmení: Radek Zatočil

Bytem: Révová 29, Brno

Narozen/a (datum a místo): 25. srpna 1984 v Brně

(dále jen „autor“)

a

### 2. Vysoké učení technické v Brně

Fakulta podnikatelská

se sídlem Kolejní 2906/4, 612 00, Brno

jejímž jménem jedná na základě písemného pověření děkanem fakulty:

Ing. Jiří Kříž, Ph.D., ředitel Ústavu informatiky

(dále jen „nabyvatel“)

## Čl. 1

### Specifikace školního díla

1. Předmětem této smlouvy je vysokoškolská kvalifikační práce (VŠKP):

☐ disertační práce

☐ diplomová práce

☒ bakalářská práce

☐ jiná práce, jejíž druh je specifikován jako .....

(dále jen VŠKP nebo dílo)

Název VŠKP: Návrh firemního databázového systému zboží

Vedoucí/ školitel VŠKP: Ing. Jiří Kříž, Ph.D.

Ústav: Ústav informatiky (UI)

Datum obhajoby VŠKP: červen 2007

VŠKP odevzdal autor nabyvateli v\*:

☒ tištěné formě — počet exemplářů 1

☒ elektronické formě — počet exemplářů 1

---

\* hodící se zaškrtněte

1. Autor prohlašuje, že vytvořil samostatnou vlastní tvůrčí činností dílo shora popsané a specifikované. Autor dále prohlašuje, že při zpracovávání díla se sám nedostal do rozporu s autorským zákonem a předpisy souvisejícími a že je dílo dílem původním.
2. Dílo je chráněno jako dílo dle autorského zákona v platném znění.
3. Autor potvrzuje, že listinná a elektronická verze díla je identická.

## **Článek 2**

### **Udělení licenčního oprávnění**

1. Autor touto smlouvou poskytuje nabyvateli oprávnění (licenci) k výkonu práva uvedené dílo nevýdělečně užít, archivovat a zpřístupnit ke studijním, výukovým a výzkumným účelům včetně pořizování výpisů, opisů a rozmnoženin.
2. Licence je poskytována celosvětově, pro celou dobu trvání autorských a majetkových práv k dílu.
3. Autor souhlasí se zveřejněním díla v databázi přístupné v mezinárodní síti
  - ☒ ihned po uzavření této smlouvy
  - ☐ 1 rok po uzavření této smlouvy
  - ☐ 3 roky po uzavření této smlouvy
  - ☐ 5 let po uzavření této smlouvy
  - ☐ 10 let po uzavření této smlouvy(z důvodu utajení v něm obsažených informací)
4. Nevýdělečné zveřejňování díla nabyvatelem v souladu s ustanovením § 47b zákona č. 111/ 1998 Sb., v platném znění, nevyžaduje licenci a nabyvatel je k němu povinen a oprávněn ze zákona.

## **Článek 3**

### **Závěrečná ustanovení**

1. Smlouva je sepsána ve třech vyhotoveních s platností originálu, přičemž po jednom vyhotovení obdrží autor a nabyvatel, další vyhotovení je vloženo do VŠKP.
2. Vztahy mezi smluvními stranami vzniklé a neupravené touto smlouvou se řídí autorským zákonem, občanským zákoníkem, vysokoškolským zákonem, zákonem o archivnictví, v platném znění a popř. dalšími právními předpisy.
3. Licenční smlouva byla uzavřena na základě svobodné a pravé vůle smluvních stran, s plným porozuměním jejímu textu i důsledkům, nikoliv v tísní a za nápadně nevýhodných podmínek.
4. Licenční smlouva nabývá platnosti a účinnosti dnem jejího podpisu oběma smluvními stranami.

V Brně dne: .....

.....  
Nabyvatel

.....  
Autor

**ABSTRAKT:**

Cílem bakalářské práce je návrh nového databázového systému pro výběr a hledání zboží. Návrh konceptuálního a logického schématu databázové struktury skladového hospodářství pro firmu. Hlavní důraz je kladen na rychlost získání informace z databáze, jednoduchost a přehlednost konečného návrhu.

**ABSTRACT:**

The purpose of this bachelor's thesis is to design and project new database system for searching and choosing the company products. Design of conceptual and logical diagram of database structure in company inventory control. Main purpose is to get the information from database as quick as possible and simplicity, transparency of database design.

**KLÍČOVÁ SLOVA**

sklad, entity, databáze, logické schéma, SQL

**KEY WORDS**

warehouse, entities, database, logical diagram, SQL

## **BIBLIOGRAFICKÁ CITACE VŠKP DLE ČSN ISO 690**

ZATOČIL, R. *Návrh firemního databázového systému zboží*. Brno: Vysoké učení technické v Brně, Fakulta podnikatelská, 2007. 59 s. Vedoucí bakalářské práce Ing. Jiří Kříž, Ph.D.

**ČESTNÉ PROHLÁŠENÍ:**

Prohlašuji, že předložená bakalářská práce je původní a zpracoval jsem ji samostatně. Prohlašuji, že citace použitých pramenů je úplná, že jsem v práci neporušil autorská práva (ve smyslu zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským, ve znění pozdějších předpisů).

V Brně dne 30.5.2007

---

podpis

## **PODĚKOVÁNÍ:**

Tímto bych chtěl poděkovat vedoucímu bakalářské práce Ing. Jiřímu Křížovi, Ph.D. za metodické a cíleně orientované vedení a rady při sepsání této práce.

Dále bych chtěl poděkovat firmě Autolak+ spol. s.r.o. za pomoc a věcné rady při zpracování této práce.



## OBSAH

<b>ÚVOD.....</b>	<b>11</b>
<b>1 VYMEZENÍ PROBLÉMU A CÍLE PRÁCE .....</b>	<b>13</b>
<b>2 ANALÝZA PROBLÉMU A SOUČASNÉ SITUACE .....</b>	<b>14</b>
2.1 PROFIL FIRMY .....	14
2.2 SWOT ANALÝZA FIRMY .....	15
2.2.1 <i>Silné stránky:</i> .....	15
2.2.2 <i>Slabé stránky:</i> .....	15
2.2.3 <i>Možnosti firmy:</i> .....	15
2.2.4 <i>Hrozby firmy:</i> .....	16
2.3 PARTNEŘI A KONKURENCE FIRMY .....	16
2.3.1 <i>Partneři a odběratelé</i> .....	16
2.3.2 <i>Konkurence firmy</i> .....	16
2.4 VYBAVENÍ FIRMY Z HLEDISKA HARDWARU A SOFTWARE .....	17
2.5 ANALÝZA SOUČASNÉHO STAVU DATABÁZE .....	17
2.6 ZÁVĚR A SHRNUTÍ ANALÝZY SOUČASNÉ SITUACE VE SPOLEČNOSTI.....	20
<b>3 TEORETICKÁ VÝCHODISKA PRÁCE .....</b>	<b>21</b>
3.1 HISTORIE SQL A DATABÁZOVÝCH SERVERŮ.....	21
3.2 DATABÁZE, SŘBD .....	22
3.3 DATABÁZOVÝ SYSTÉM .....	23
3.4 VZTAHY MEZI ENTITAMI .....	24
3.5 NORMALIZACE RELACÍ .....	25
3.6 JAZYK SQL A JEHO PODMNOŽINY.....	27
3.7 MOŽNOSTI VÝBĚRU SQL SERVERU.....	29
3.7.1 <i>Komerční SQL servery</i> .....	30
3.7.2 <i>Open Source SQL servery</i> .....	31
<b>4 VLASTNÍ NÁVRHY ŘEŠENÍ .....</b>	<b>33</b>
4.1 SEZNAM POŽADAVKŮ FIRMY .....	34
4.2 KONCEPTUÁLNÍ NÁVRH .....	34
4.2.1 <i>Popis entit a vazeb v ER diagramu</i> .....	36
4.3 LOGICKÉ SCHÉMA.....	37
4.3.1 <i>Popis atributů entit</i> .....	39
4.4 NÁVRH TABULEK A NORMÁLNÍ FORMY .....	43
4.5 NÁVRH SOFTWAREOVÉHO VYBAVENÍ .....	44
<b>5 PŘÍNOS (EFEKTIVNOST) NÁVRHŮ ŘEŠENÍ.....</b>	<b>46</b>

<b>6</b>	<b>ZÁVĚR.....</b>	<b>47</b>
	<b>SEZNAM OBRÁZKŮ A TABULEK .....</b>	<b>48</b>
	<b>SEZNAM POUŽITÉ LITERATURY.....</b>	<b>49</b>
	<b>SEZNAM POUŽITÝCH ZKRATEK.....</b>	<b>51</b>
	<b>PŘÍLOHY .....</b>	<b>52</b>

# Úvod

*„Technicky se dá udělat mnoho věcí, ale jen ekonomicky přijatelné projekty se dočkají uskutečnění.“<sup>1</sup>*

*Rober Noyce (Intel)*

Databázové systémy jsou jeden z prostředků, bez kterých bychom si v posledních desetiletích nedokázali představit existenci a činnost firmy či společnosti. Umožňují firmám relativně levně a efektivně spravovat veškerá svá data, počínaje informacemi o svých zákaznících, zboží, dodavatelích a účetnictví až po detailní informace o svých nákladech, výnosech, skladových zásobách atd. Proto je na databáze upřen velký zájem odborné veřejnosti ale i běžných uživatelů PC a vzniká velký počet projektů, které mají vývoj databází usnadnit či nějakým způsobem inovovat. Existuje celá řada sofistikovaných publikací, které přinášejí detailní informace jak vytvořit a navrhnout optimalizovanou databázi. Při realizaci návrhu databázového systému můžeme využít komerčních databázových serverů mezi které patří Microsoft SQL Server, Oracle, Informix, InterBase a mnoho dalších tak i bezplatné servery MySQL či Oracle Database XE, které zajišťují veškerou správu dat.

Jak naznačil úvodní citát, v dnešní době je zapotřebí tvořit projekty, hodnoty a například právě návrhy databázových systémů, které jsou spojeny s výrazem ekonomická přijatelnost či ekonomická prospěšnost. V širším významu můžeme výraz „ekonomická přijatelnost“ chápat jako snahu o nalezení nejvhodnějších postupů a technik k vyřešení dané situace. Dokonce ekonomie je někdy definována jako „věda o rozhodování“.

---

<sup>1</sup> EUROEKONOM. *Svět ekonomie, obchodu, investic a liberalismu v citátech*. [online]. Poslední aktualizace 15.03.2007 [cit. 2007-05-22]. Dostupný z WWW:< <http://www.euroekonom.cz/citaty.html>>.

Při zpracování bakalářské práce jsem usiloval o to aby můj navrhovaný databázový systém byl ekonomicky nákladově přijatelný a ekonomicky výnosově prospěšný. Zadavateli by měl přinést ekonomické zhodnocení minimalizováním pořizovacích nákladů na straně jedné a zvýšením ekonomické výkonnosti firmy např. větším počtem obslužených a spokojených zákazníků za stejný časový úsek či ušetřeným časem při práci se systémem, neboť čas jsou peníze, na straně druhé.

# 1 Vymezení problému a cíle práce

Firma Autolak+ spol. s.r.o. se potýká se závažným problémem s databází sortimentu produktů. Databáze sortimentu zboží je vytvořena pouze na základě specifických katalogových čísel zboží, převzatých z dodavatelských katalogů. Při prodeji zboží nastává problém, kdy je nezbytné příslušné zboží najít v konkrétním katalogu příslušného výrobce či dodavatele a do PC databáze zadat katalogové číslo.

Tato metoda se ukazuje jako velmi zdoluhavá a neoperativní a také vyžaduje o jednu pracovní sílu více – fakturanta. S tímto problémem se setkávají zaměstnanci. Problém je velmi naléhavý, neboť se s databází pracuje každý den a je to nedílná součást pracovního procesu.

Cílem je návrh zcela nového databázového systému, který umožní zaměstnancům intuitivně vyhledávat v databázi produktů, nebude zapotřebí vyhledání identifikačního čísla produktu v katalogu od výrobce. Dílčí výsledky této nové databáze jsou, výrazné zrychlení celého procesu při výdeji zboží zákazníkovi a také možnost odbourání jedné pracovní síly v podniku, což by výrazně snížilo náklady na provoz firmy a v neposlední řadě je to také zlepšení kontroly při přípravě zákaznické objednávky a usnadnění optimalizace skladových zásob.

Pro uskutečnění stanovených cílů je třeba získat veškerá data o zboží a odběratelích ve firmě. Je zcela nezbytné vlastnit informace o tom, které charakteristiky zboží jsou požadovány v databázi. Dalším bodem je průzkum firemního softwarového vybavení. Jaké možnosti jsou pro implementaci nové databáze, přitom jednou z priorit jsou co nejmenší náklady na uskutečnění navrhovaného řešení. Po tomto průzkumu je možné navrhnout datovou strukturu a relace v SQL jazyku na konceptuální úrovni a následně logické úrovni.

## **2 Analýza problému a současné situace**

### **2.1 Profil firmy**

Společnost Autolak+ spol. s.r.o. byla založena 30. září 1991 podpisem společenské smlouvy dvou společníků. Vklady byly v peněžitě formě a to každý ze společníků vložil 500 000 Kč, tedy základní kapitál společnosti je 1 000 000 Kč. Jedná se o malý podnik s méně jak 10 zaměstnanci.

Firma Autolak+ spol. s.r.o. se již od začátku svého podnikání, se stala jedním z mála distributorů autolaků Glasurit na jižní Moravě. Postupně se vypracovala na výhradního zástupce pro jižní Moravu generálního dovozce autolaků Glasurit do ČR firmy Autop Bohemia. V roce 1999 se firma po dohodě s generálním dovozcem stala také distributorem autolaků Glasurit pro severní Moravu.

Podstatou činnosti je bezplatné zapůjčení míchacího zařízení na autolaky do autoservisů a následné dodávky lakovacích materiálů. Firma tedy získává síť stálých odběratelů, které dlouhodobě zásobuje materiálem a poskytuje jim know-how formou školení zaměstnanců ve spolupráci s firmou Autop Bohemia a výrobcem laků BASF coating a.g. Dále firma zajišťuje poradenství při využívání internetové databáze odstínů barev a s obsluhováním databáze receptur odstínů laků instalované do PC. Nezbytnou službou v konkurenčním boji je také zásobováním veškerým technickým a spotřebním zbožím pro lakovny a operativní záruční i pozáruční opravy technického zboží. K těmto velkoobchodním činnostem provozuje firma i maloobchod autolaků a technického zboží. Prodej technických materiálů (např. stříkací kabiny, infrazářiče, brusiva, průmyslové vysavače, maskovací materiály atd.) je pojat jako podpůrný program hlavního nosného produktu – distribuce laků a barev, ale jeho obrat činí cca. 30 % celkového prodeje zboží.

K Obchodnímu sortimentu který je mimo autolaky firmy Glasurit prodáván a distribuován patří např. produkty firem SATA, BASF salcomix, DeVILBISS, KLINGSPOR, 3M, FESTOOL (FESTO), RUPES, HAMACH. a dalších firem zabývajících se prodejem stříkacích pistolí, materiálu, krycích oděvů, brusných zařízení

atd. Obrat firmy je ze 73 % realizován stalými zákazníky - tj. autoservisy - lakovny, kterým bylo bezplatně zapůjčeno a nainstalováno míchací zařízení laků a s nimiž jsou smluvně specifikovány obchodní a dodavatelské vztahy pro trvalé zásobování materiály Glasurit. Těchto smluvních vztahu je třicet.

## **2.2 SWOT analýza firmy**

### **2.2.1 Silné stránky:**

- špičkový produkt s výbornou konkurenceschopností,
- výborný obchodně technický servis,
- vysoká odborná způsobilost zaměstnanců,
- dlouhodobě stabilizovaná odběratelská a zákaznická síť.

### **2.2.2 Slabé stránky:**

- nákup stěžejního obchodního produktu pouze od autorizovaného importéra do ČR; nikoliv od výrobce,
- obchodní strategie je silně závislá na podmínkách vytvořených importérem,
- vzhledem k existenci importéra do ČR, má firma relativně nízkou ziskovost,
- maximálně vyhrocené obchodně konkurenční prostředí v dané obchodní komoditě – konkurenční firmy mají možnost přímého importu produktu a jsou mnohonásobně kapitálově silnější,
- absence prezentace firmy Autolak+ spol. s.r.o. v síti internet,
- nedostačující a zastaralý databázový systém pro správu zboží ve skladu.

### **2.2.3 Možnosti firmy:**

- udržování stále zákaznické a odběratelské sítě,
- vytvoření WWW stránek s prezentací zaměřenou na profesionální lakovací servisy,
- návrh a implementace nového moderního IS,
- sloučení s importérem nosného produktu,
- možnost získání práv pro přímý import,

- výhodnější obchodní podmínky při zajišťování produktu od importéra do České republiky.

#### **2.2.4 Hrozby firmy:**

- Vzhledem k tomu, že nenastaly poslední tři zmiňované body, firma se dostala do vážných problémů v boji o udržení podílu na trhu. Jak již bylo zmíněno, nepříznivě stanovené podmínky importérem a vyhocené konkurenční prostředí vede ke krizi.

### **2.3 Partneři a konkurence firmy**

#### **2.3.1 Partneři a odběratelé**

Mezi stálé zákazníky a odběratele produktů společnosti Autolak+ spol. s.r.o. můžeme zařadit významné značky a autoservisy po celé severní a jižní Moravě, k nejvýznamnějším odběratelům patří:

AVIA Ivančice (IVACAR), Spofort Brno, Ford Rašino Brno, Autonova Brno, Mercedes Hošek Brno, Autolakovna Tauš Vyškov a mnoho dalších meších autolakoven a servisů. Průměrný měsíční odběr těmito autolakovnami se pohybuje v rozpětí 100 000 – 500 000 Kč.

Zbýlých 27 % z celkového objemu obrátu je realizován přímým prodejem za hotové, přičemž minimálně polovinu tohoto objemu tvoří stálí zákazníci z řad soukromých podnikatelů – autolakýrníků, reklamních agentur ( Snip a CO. ) a výrobních organizací ( Manex, Industrial, Benč).

#### **2.3.2 Konkurence firmy**

Mezi konkurenční značky autolaků, které jsou v České republice distribuovány patří Herberts Standox, PPG, ICI, Spies Hecker, Dupot, Sikens, RM, De Beer, Max Mayer. Podíl společnosti distribuující autolaky značky Glasurit na území jižní a severní Moravy v daném oboru je zhruba 35 %, dále pak následují konkurenti s významnějším podílem, kteří distribuují značky Herbert Standox cca. 21 % zmíněného trhu a PPG s cca. 13 % podílem.



## **2.4 Vybavení firmy z hlediska hardwaru a softwaru**

Firma vlastní sedm osobních počítačů. Ve firmě se nachází jeden serverový PC určený pro zálohu citlivých účetních údajů a transakcí. Dále jsou PC používány při prodeji zboží k provozu databáze skladu a databáze vzorníků barevných odstínů.

Nároky na hardwarové vybavení těchto počítačů nejsou příliš velké, protože jsou zejména používány k provozu katalogů barevných odstínů, katalogu systému míchaní a k účetním programům, textovým editorům a komunikačním programům. Tyto programy jsou většinou tvořeny na bázi systému MS-DOS. Převážně se jedná o strašné počítače bez grafických karet s menší kapacitou pevných disků do 40 GB a nepříliš velkou kapacitou operační paměti. Samozřejmě je instalovaná síťová karta, protože jsou firemní počítače připojeny do sítě Internet a také ve firmě funguje vlastní LAN. Kvůli těmto hardwarovým omezením je zapotřebí vybrat správný SQL server tak, aby náklady na jeho pořízení byly co nejmenší a firma nemusela investovat do nového technického vybavení popřípadě i nového softwaru.

Firma vlastní licence k softwarovým produktům Microsoft Windows 98, která jsou nainstalovány na prodejních PC a dále pak licenci k Windows XP, který je nainstalován na serveru společnosti.

## **2.5 Analýza současného stavu databáze**

Analýza současného stavu ve společnosti Autolak+ spol. s r.o. byla vytvořena na základě jednání s vrcholovým managementem společnosti, koncovými uživateli dané stávající databáze, tedy pracovníky, kteří prodávají zboží zákazníkům i osobními zkušenostmi při práci s touto databází při výkonu odborné praxe ve firmě. Přehledné shrnutí závažných chyb, nedostatků v současném databázovém systému:

### **Vyhledávání sortimentu zboží**

Tento nedostatek patří k jednomu z nejzávažnějších chyb ve stávající databázi. Vyhledání příslušného zboží je umožněno jen na základě dlouhého katalogového čísla,

které není pouze numerické ale je složeno z kombinace čísel a písmen. Prodávající pracovník musí na základě zákaznického požadavku vyhledat ve velkém množství katalogů od různých dodavatelů, mnohdy shodné zboží a poté může provést cenové, funkční srovnání a vybrané identifikační číslo zboží zadat do databázového programu. Všeobecně stávající databáze nepodporuje vyhledávání v sortimentu zboží. Vznese-li zákazník požadavek na cenu určitého výrobku či kolik kusů výrobku může dostat, vše musí být pracovníkem zdlouhavě hledáno v katalogu a poté na základě zadání příslušných katalogových čísel zjištěno ze skladového hospodářství.

### **Identifikace pracovníka**

Databáze neumožňuje identifikovat pracovníka, který vytvořil dodací list. Pro vstup do databáze neexistují žádné restriktce, přihlašovací login a heslo atd. je otevřená a pracuje s ní jakýkoliv zaměstnanec. Jedná se o závažný problém absence systému práv, díky tomuto nedostatku mohou vzniknout rozsáhlé ekonomické ztráty společnosti. S tímto problémem souvisí i další nedostatek a to ochrana informací. Databáze zboží je zpřístupněna komukoliv a mohou být z ní zjištěny důležité ekonomické údaje.

### **Nemožnost propojení s účetním software**

Dalším nedostatkem je osamocená funkce databázového programu. Je zapotřebí aby databáze skladu byla přímo propojena s účetním software. Informační struktura ve společnosti je decentralizovaná. Existují jednotlivé informační buňky – účetní, fakturant, sklad, které mezi sebou špatně kooperují a předávají si data neefektivním způsobem. Ke zlepšení tohoto stavu by bylo zapotřebí zcela nového návrhu IS a jeho implementaci pro firmu.

### **Výdej zboží**

Výdej zboží je uskutečněn pomocí dodacího listu, přičemž je proces zaúčtování vydaného zboží výrazně zpomalen, protože vyplněný dodací list musí být doručen na fakturační oddělení, kde je na základě tohoto listu vystavena faktura. Databázový

system není propojen s účetním software popřípadě není zahrnuta funkce vystavení faktury. Na dodacím listu nejsou všechny potřebné informace ale pouze ID produktu, počet kusů, cena bez DPH za kus a celková cena bez DPH, adresa dodavatele IČ, DIČ, adresa odběratele, IČ, DIČ a datum výdeje zboží. S tímto problémem souvisí i nemožnost nastavit specifickou odběratelskou adresu v případě že, zboží je doručeno na jinou adresu než je adresa fakturační, tedy adresu tzv. pobočného závodu.

### **Lidský faktor**

V případě, kdy pracovník odpovědný za stav skladových zásob zapomene předat vystavený a potvrzený dodací list k fakturaci dochází k finančním ztrátám za vydané zboží, které není fakturováno. Při následném zjištění těchto chyb jsou případné opožděné fakturace přijímány odběrateli velmi negativně. Zpožděná fakturace vydaného zboží vytváří obchodní potíže odběrateli při následné obchodní i účetní realizaci tohoto zboží. Se zbožím, které není zaúčtováno na podkladu řádné faktury nelze obchodně manipulovat. Tento problém je velmi častý a je nezbytné, aby byl vhodným vytvořením databáze odstraněn.

### **Optimalizace skladu**

Stávající databáze zboží neumožňuje vytvořit informace o minimálním nebo naopak maximálním stavu zásob ve skladu a nedokáže vytvořit statistické informace o prodejní obrátkovosti jednotlivých skladových položek. Kvůli absenci těchto základních skladových údajů mohou opět vznikat značné ekonomické ztráty.

### **Tvorba dokladů**

Nainstalovaná databáze neumožňuje taktéž tvorbu důležitých a nezbytných průvodních dokladů k prodanému zboží jako jsou nákladové listy a bezpečnostní doklady s údaji o hmotnosti, objemu, hořlavosti, obsahu nebezpečných chemických látek a dalších dokumentů.

## **Srovnávací statistiky**

Databáze nepodporuje jakékoliv zobrazení statistik. Společnost by potřebovala zjistit statistiky prodejnosti u téměř charakterově shodných výrobků od různých dodavatelů.

## **2.6 Závěr a shrnutí analýzy současné situace ve společnosti**

Proces výdeje zboží, fakturace i účtování je ve společnosti Autolak+ spol. s.r.o. je zastaralý a velmi neefektivní. V dnešní době moderní výpočetní techniky, kvalitních databázových serverů je tento koncept prodeje zboží již ojedinělý.

Hlavní příčinou zpomalení a neefektivnosti je špatně navržené databázové schéma skladu, které prakticky nepodporuje jakoukoliv komunikaci s uživatelem.

Nejzásadnější chybou staré databáze je, že umožňuje pouze vyhledávání podle jediného kritéria a tím je katalogové číslo, které se musí většinou nejprve vyhledat v katalogu. Toto číslo je obtížné k zapamatování, i pro pracovníky s dlouholetou praxí ve společnosti.

Při hledání řešení problému se starou databází při vyhledávání produktu bylo potřeba analyzovat jednotlivé skupiny zboží od různých výrobců. Výsledkem bylo zjištění, že zboží od různých výrobců je možno kategorizovat do stejných skupin a tím velmi zjednodušit způsob vyhledávání zboží. Výsledky budou popsány v kapitole Vlastní návrhy řešení.

## 3 Teoretická východiska práce

### 3.1 Historie SQL a databázových serverů

Jazyk SQL (Structured Query Language) vznikl v 70. letech 20. století. V té době byl však pojmenován SEQUEL (Structured English Query Language). Cílem bylo vytvořit jazyk, ve kterém by se příkazy tvořily syntakticky co nejbližší přirozenému jazyku, tedy angličtině. Jako první byl použit v Systému R, vyvinutém v kalifornské laboratoři IBM. Od té doby se jazyk rozšířil a byl používán v ostatních systémech. K vývoji jazyka se přidaly další firmy. Relační databáze byly stále významnější, a bylo nutné jejich jazyk standardizovat. Americký institut ANSI původně chtěl vydat jako standard zcela nový jazyk RDL. SQL se však prosadil jako standard a ANSI založil nový standard na tomto jazyku. Tento standard bývá označován jako SQL-86 podle roku, kdy byl přijat a v roce 1987 získal jazyk certifikát ISO. V dalších letech se ukázalo, že SQL-86 obsahuje některé nedostatky a naopak v něm nejsou obsaženy některé důležité prvky týkající se hlavně integrity databáze. Výsledná zpráva byla zveřejněna v roce 1989 organizací ISO. Tomuto rozšíření se říká SQL89. V roce 1992 byl proto přijat nový standard SQL-92 (někdy se uvádí jen SQL2). Zatím nejnovějším standardem je SQL3 (SQL-99), který reaguje na potřeby nejmodernějších databází s objektovými prvky.<sup>2</sup>

První významnější produkty, kde byl jazyk SQL zabudován, byly od IBM: DB2, SQL/DS (oba vycházejí ze Systému R). S nástupem počítačů třídy PC začaly i ostatní výrobci více uvažovat o SQL. Asi nejznámější je firma Ashton-Tate se svou dBASE IV (i když implementace SQL nebyla úplná). Jako další jmenujme INFORMIX-SQL, INGRES pro PC, ORACLE, SQLBase, XDB II, XQL. Postupem času se začal jazyk SQL přenášet i na systém UNIX a od jedinouživatelských úloh se přecházelo k serverům založených na SQL. Jmenujme aspoň SŘBD ORACLE, INFORMIX, PROGRESS a INGRES. Dále např. SQL Server od firmy Sybase či Microsoft.<sup>3</sup>

---

<sup>2</sup> KOCAN, M. *Víte, co je SQL?* [online]. c2007, poslední aktualizace 26.10. 1998 [cit. 2007-03-20]. Dostupný z WWW: <<http://www.zive.cz/h/Programovani/AR.asp?ARI=3320>>.

<sup>3</sup> ŽÁK, K. *Historie relačních databází*. [online]. c1998-2007, poslední aktualizace 19.10. 2001 [cit. 2007-03-15]. Dostupný z WWW: <<http://www.root.cz/clanky/historie-relacnich-databazi/>>.

## **3.2 Databáze, SŘBD**

Databázi si můžeme představit jako úložiště údajů, které jsou uloženy a zpracovávány nezávisle na aplikačních programech. V databázi jsou uchovány jak vlastní údaje, ale i relační vztahy mezi jednotlivými prvky a objekty v databázi, schémata popisující struktury údajů a doménová, integritní omezení.

Přístup k údajům uloženým v databázi obstarává objekt databáze, který je uváděn pod zkratkou SŘDB – systém řízení báze dat, nebo anglicky database management systém (DBMS). Díky tomuto software uživatel nemusí znát fyzickou strukturu uložení údajů, protože k údajům v databázi přistupují prostřednictvím systému řízení báze dat. Komunikace klienta nebo aplikačního programu SŘBD probíhá pomocí jazyka SQL. Většina SŘBD v současných programech má charakteristické standardní rysy, jako:

### **Transakční zpracování**

To znamená, že složitější manipulace s údaji, která se skládá z posloupnosti určitých kroků, je vykonávána jako transakce, která převede databázi z jednoho konzistentního stavu do druhého. Jednodušeji lze říci, že buď všechny operace v transakci proběhnou úspěšně, nebo neproběhnou vůbec, tedy databáze bude uvedena do stavu v jakém byla před transakcí.

### **Zotavení se z chyb**

Když při zpracování údajů dojde k chybě nebo při složitých transakcích, systém se z této chyby snaží zotavit. Když vlivem nějaké nepředvídatelné události, například zhroutení systému nebo živelné pohromy, dojde ke ztrátě údajů, měl by se systém po těchto událostech zotavit a pokračovat v práci například se záložní databází.

### **Víceuživatelský přístup**

Předpokládá se efektivní řízení přístupu k údajům ze strany více uživatelů nebo klientských aplikací. SŘBD proto musí umět definovat přístupová práva jednotlivých uživatelů k databázovým objektům a specifikovat rozsah jejich oprávnění. Někteří

uživatelé mohou údaje do databáze zapisovat, případně mazat, jiní mají zpřístupněné jen jejich čtení.

### **Ochrana údajů**

Pojem ochrana údajů lze definovat jako ochranu před ztrátou údajů, například při haváriích hardwaru, ale můžeme ho také definovat jako ochranu údajů před jejich možnou krádeží, nebo zneužitím. Také jako nutnost zabránění neoprávněných osob k údajům v databázích.<sup>4</sup>

### **Architektura**

Záleží na velikosti a objemu dat, která mají být zpracovávána a uchovávána. Data mohou být uložena centralizovaně na jednom serveru, nebo distribuovaně na více serverech. O distribuovaném zpracování dat hovoříme především v případech splnění několika základních podmínek, mezi které patří například geografické oddělení databázových serverů – a to i od různých firem. Podstatné také je, že uživatele nemusí zajímat, která část systému jeho požadavek zpracovává či kde jsou data uložena.

## **3.3 Databázový systém**

Pojem databázový systém zahrnuje jednak údaje, které jsou uloženy a spravovány v databázi, ale i software pro přístup k těmto údajům. Souhrn nástrojů, postupů a technik, které se využívají v souvislosti s databázemi, nazýváme databázová technologie. Databázový systém je tedy tvořen systémem řízení báze dat SŘBD a databází samotnou. Databázové systémy rozdělujeme na několik základních typů:

- **hierarchické a síťové** – u těchto typů databázových systémů jsou aplikační programy závislé na databázi, z čehož vyplývá například problematická údržba,
- **relační** – pro tyto databáze je typická neprocedurální manipulace s daty, ukládání jednoduchých dat s pevnou strukturou, tedy v tabulkové formě,

---

<sup>4</sup> LACKO, L. *SQL : Hotová řešení*. 1. vyd. Brno : Computer Press, 2003. s. 21. ISBN 80-7226-975-5.

- **objektové** – databázové systémy používají složité datové struktury a složitá pravidla založená na obchodní logice známe pod anglickým názvem Business rules.<sup>5</sup>

Převážná většina SRDB při uspořádání dat používá relační model dat. V tomto modelu jsou data uspořádána do tabulek. Tabulka (entita) většinou shromažďuje informace o jednom druhu objektu. Sloupce tabulky se nazývají položky nebo atributy, řádkům říkáme záznamy. Každá tabulka má své jedinečné jméno. Každý záznam v tabulce musí být nějak identifikován. K tomu slouží jedinečný klíč, zvaný primární klíč. Každý atribut má svůj typ – celé číslo, reálné číslo, datum/čas, řetězec, atd.<sup>6</sup>

Relační model databáze má tyto podmínky, které byly formulovány Dr. E. F. Coddem :

- všechny údaje v databázi jsou uloženy v tabulkách,
- fyzická struktura údajů a jejich uložení je nezávislé a úplně od uživatelů odstíněné, to znamená, že neexistují nějaké uživateli viditelné přístupové cesty,
- předpokládáme existenci databázového jazyku, který umožňuje realizovat minimální operace selekce, restrikce, projekce a spojení.<sup>7</sup>

### 3.4 Vztahy mezi entitami

Pro lepší představu o vztazích, které mohou nastat mezi jednotlivými tabulkami uvedu jednoduchý příklad z praxe. Firma dodává zboží mnoha zákazníkům a chce mít o každém zákazníkovi kontaktní údaje, jako telefonní číslo, adresa, jméno atd. Zákazník ale neodebírá pouze jednu položku, ale odebírá více položek zboží, tedy má více záznamů v tabulce. Bylo by nelogické a zdlouhavé pořád dokola opakovat stejné informace u každého záznamu znova. Proto se dají údaje o zákazníkovi do speciální

---

<sup>5</sup> LACKO, L. *SQL : Hotová řešení*. Brno : Computer Press, 2003. s. 21. ISBN 80-7226-975-5.

<sup>6</sup> KOSEK, J. *PHP: tvorba interaktivních internetových aplikací*. Grada Publishing, 1998. s. 123-125. ISBN 80-7169-373-1.

<sup>7</sup> LACKO, L. *SQL : Hotová řešení*. Brno : Computer Press, 2003. s. 29. ISBN 80-7226-975-5.



tabulky. Tyto problémy se řeší pomocí tzv. vztahů mezi tabulkami. Tvoříme tzv. relace, neboli spojení jednotlivých atributů tabulek.

Relace mezi tabulkami vlastně popisují vztahy mezi objekty reálného světa, které tyto tabulky představují. Při návrhu databázových tabulek, na které navazuje aplikační logika, můžeme definovat tři základní druhy vztahů:

**1 : 1** - první entitě odpovídá maximálně jedna druhá entita.

**1 : N** - první entitě odpovídá více druhých entit. Ale druhé entitě odpovídá pouze jediná entita.

**M : N** - první entitě odpovídá více druhých entit. A také i naopak, druhé entitě odpovídá více prvních entit.

Přičemž většina databázových systémů nedokáže přímo pracovat se vztahy typu  $M : N$ , v praxi se použije dekompozice, tedy tento vztah se rozloží na dva vztahy typu  $1 : N$  pomocí „propojovací“ tabulky.

Unární relace – v praxi se vyskytuje ještě jeden typ relace, a to relace jedné tabulky sama se sebou. Nejvyšší hierarchická úroveň je v takovéto vazbě tvořena jedním prvkem, který je svázaný s prvky o jednu úroveň níže atd. Tento vztah se často označuje jako nadřazený – podřazený.<sup>8</sup>

### **3.5 Normalizace relací**

Co je to normalizace? Normalizace je odstranění redundantních dat, omezení složitosti (rozložení složité relace na dvojrozměrné tabulky) a zabránění tzv. aktualizacím anomáliím (např. abychom smazáním všech knih autora nepřišli o data o autorovi). Což by mělo vést k databázi přehlednější, rozšiřitelnější a výkonnější.

Normalizace by měla vést k vzniku tabulek, které lze snadno udržovat a efektivně se na

---

<sup>8</sup> LACKO, L. *SQL : Hotová řešení*. Brno : Computer Press, 2003. s. 36. ISBN 80-7226-975-5.

ně dotazovat. Normalizované schéma musí zachovat všechny závislosti původního schémat a relace musí zachovat původní data, což znamená, že se musíme pomocí přirozeného spojení dostat k původním datům.

### **První normální forma (1NF)**

Relace je v první normální formě, pokud každý její atribut obsahuje jen atomické hodnoty. Tedy hodnoty z pohledu databáze již dále nedělitelné.

**Tabulka č. 1 - Nesplnění 1NF**

Jméno	Příjmení	Adresa	Telefony
Petr	Novák	Valtická 8 Brno	602555669;444666333
Luděk	Starý	Mokrá 5 Praha	369222335;446688992

Aby tabulka byla v 1NF musíme oddělit telefonní čísla do samostatné tabulky a rozdělit atribut adresa.

### **Druhá normální forma (2NF)**

Platí pouze tabulky, které mají více primárních klíčů. Relace se nachází v druhé normální formě, je v první normální formě a každý neklíčový atribut je plně závislý na primárním klíči, a to na celém klíči a nejen na nějaké jeho podmnožině.

### **Třetí normální forma (3NF)**

Je splněna, když je splněna 2NF a zároveň neexistuje závislost mezi jednotlivými atributy tabulky. Pokud vezmeme tabulku zobrazenou níže, tak vidíme jasnou závislost mezi atributem PSČ a atributem Město. Proto je lépe přidat tabulku, kde u identifikačního čísla města bude zapsáno i příslušné PSČ. Na druhou stranu není vždy výhodné přesně se držet této formy, jelikož to může vést k jistým komplikacím např. při tvorbě dotazů.

**Tabulka č. 2 - 3NF, kdy opravdu normalizovat?**

r. č	Jméno	Příjmení	Město	PSČ	Funkce	Plat
8005253233	Petr	Novák	Brno	62701	CEO	30000
7505154313	Luděk	Starý	Praha	10110	Programátor	25000
7903234321	Aleš	Dobrý	Plzeň	36598	Účetní	20000

V některých případech vede striktní dodržování 3 NF k zbytečně složitým a nepřehledným tabulkám a někdy i ke ztrátě výkonu. Je zbytečné vytvářet další tabulku v databázi s několika tisíci PSČ v České Republice, když ve firmě pracuje pouze 10 zaměstnanců.

#### **Čtvrtá normální forma (4NF)**

Platí pokud tabulka popisuje jeden objekt nebo událost. Tedy, tabulka s atributy id, zboží, dodavatel, kusu, adresa nesplňuje 4NF (kvůli adrese – popisujeme odběr zboží).

#### **Pátá normální forma (5NF)**

Platí pokud nelze do tabulky přidat jeden či více sloupců tak, aby se tabulka nerozpadla na další tabulky.<sup>9</sup>

### **3.6 Jazyk SQL a jeho podmnožiny**

Jazyk SQL můžeme použít jako dotazovací jazyk pro práci s údaji v relační databázi, ale také jako část hostitelského jazyka pro vývoj databázových aplikací. Větší a složitější databáze potřebují více než jenom základní příkazy. Proto se vyskytly různé nástavby klasického SQL. Součástí SQL serveru od firmy Microsoft je také jazyk Transact – SQL, který umožňuje používat SQL jako standardní programovací jazyk, pomocí něhož lze definovat přesnou programovou logiku. U databázové platformy Oracle můžeme uvést implementovaný procedurální jazyk PL/SQL. Obecně můžeme

---

<sup>9</sup> *Teorie relačních databází : Normalizace.* [online]. c2005-2007, poslední aktualizace 19.5. 2006 [cit. 2007-03-24]. Dostupný z WWW: <<http://www.manualy.net/article.php?articleID=13>>.

jazyk SQL rozdělit na dvě základní množiny:

### **Data Definition Language ( DDL )**

V českém jazyce též znám pod názvem Jazyk pro definici dat ( JDD ). Tato část jazyka SQL umožňuje definovat struktury a vytvářet v databázi objekty, jako tabulky, pohledy, indexy, sekvence a podobně, případně měnit jejich strukturu nebo je odstraňovat. Je třeba rozlišit rozdíl mezi vymazáním a zrušením. Vymazání patří do druhé množiny jazyka SQL, údaje z tabulky můžeme vymazat ale tabulka jako objekt zůstane zachována, zrušíme-li tabulku přijdeme o veškerá data v ní uložená i o samotnou tabulku v databázi. Do této skupiny patří například často využívané příkazy: CREATE DATABASE, CREATE TABLE, DROP TABLE, ALTER TABLE, CREATE INDEX, CREATE VIEW, DROP INDEX, ALTER SEQUENCE atd.

### **Data Manipulation Language ( DML )**

Tato část jazyka SQL, známá v češtině jako Jazyk pro manipulaci s daty (JMD), umožňuje manipulaci s údaji jako je například výběr a vkládání údajů a jejich aktualizaci, vymazávání údajů a samozřejmě také jeden z nejpoužívanějších příkazů pro práci s databází SELECT pro výběr údajů. Do této skupiny patří čtyři hlavní příkazy: SELECT, INSERT, UPDATE a DELETE

Kromě těchto dvou hlavních množin jazyka SQL můžeme ještě definovat dvě neméně významné podmnožiny tohoto jazyka a to:

### **Data Control Language ( DCL )**

Tato část jazyka SQL obsahuje speciální příkazy pro řízení provozu a údržbu databáze. Také můžeme přidělovat a odebírat uživatelská privilegia jednotlivým uživatelům a skupinám uživatelů:

CREATE USER, DROP USER, ALTER USER, GRANT, REVOKE

### **Transaction Control Commands ( TCC )**

Poslední podmnožina jak již název napovídá slouží pro řízení transakcí. Do této skupiny patří například příkazy:

### 3.7 Možnosti výběru SQL serveru

Výběr SQL serveru je klíčový. Před samotným navržením datové struktury, relací a dalších nezbytných prvků databáze musíme vybrat právě vhodný databázový server. Existuje mnoho kritérií na základě kterých se rozhodujeme, který databázový server zvolíme. Je-li jedním z hlavních kritérií cena pak se nabízejí dvě možnosti, jednou z nich je komerční software, který je ve vyšší cenové hladině ale za tuto vyšší cenu dostaneme poměrně spolehlivý produkt, kvalitní servis, dokumentaci i zákaznickou podporu. Proto instituce jako pojišťovny, banky, státní úřady a jiné raději sáhnou po dražším produktu u kterého se mohou spolehnout na profesionální servis v případě výskytu problému. Naopak potřebujeme-li vytvořit menší databázi pro rodinou firmu, kde potřebujeme minimalizovat náklady na výstavbu databáze, zvolíme Open Source databáze, kde cena za pořízení je minimální v případě komerčního využití nebo dokonce žádná. Za nízké pořizovací náklady ale nemůžeme spoléhat na příliš kvalitní servis a zákaznickou podporu.

Dalším z kritérií, které je neméně důležité je kompatibilita s ostatními programy a na jakém operačním systému chceme databázový server provozovat.

Při výběru samozřejmě musíme hodnotit i výkon databázového serveru, záleží nám na tom, aby data byla okamžitě dostupná bez dlouhého čekání. Proto vytváříme-li databázi s velkým množstvím entit a záznamů je naší jasnou prioritou výkon databázového serveru.

Víme-li, že s databází bude pracováno v České republice je potřeba brát v úvahu podpora češtiny, řazení záznamů podle české abecedy a další specifika, kterými se vyznačuje český jazyk.

Důležitým prvkem při rozhodování je také, v jakém prostředí chceme data

---

<sup>10</sup> LACKO, L. *SQL : Hotová řešení*. Brno : Computer Press, 2003. s. 23. ISBN 80-7226-975-5.

z databáze prezentovat a zda potřebujeme vytvořit výslednou aplikaci, která bude obsluhovat danou databázi, může se jednat o GUI navržené v jazyku C++ nebo C# nebo o jednoduché internetové stránky s formuláři zpracované v skriptovacím jazyk PHP, Java atd.

Jako jedno z mnoha kritérií pro výběr SQL serveru bych uvedl podporu funkcí jako jsou trigger, vnořené dotazy, náhledy, definice vlastních datových typů.

Kategorie, která nesmí zůstat opomenuta je hardwarová náročnost jednotlivých SQL serverů. Komerční řešení nabízejí pěkné grafické uživatelské rozhraní avšak jejich náročnost na velikost operační paměti, výkon procesoru a kapacitu pevných disků bývá mnohonásobně větší než u Open Source databází.

Kategorií podle kterých je možno vybrat SQL server je opravdu mnoho, v následující kapitole uvedu několik SQL serverů, rozdělené na komerční a Open Source řešení s posledními produkty dostupnými na trhu, cenovou dostupností a budoucím vývojem.

### **3.7.1 Komerční SQL servery**

#### **Oracle**

Poslední verzi databázového serveru od této společnosti je Oracle Database 10g. Tento typ SQL serveru je dostupný ve čtyřech verzích. Kde verze XE (Express Edition), poprvé vydána v roce 2005, je základní verze volně ke stažení. Je určena pro malé firmy, začínající vývojáře a administrátory. I tato edice však již obsahuje řadu pokročilých funkcí databáze Oracle, jako je například podpora XML, fulltextového vyhledávání nebo práce s prostorovými daty. Oracle Database XE je k dispozici pro 32bitové operační systémy Linux a Windows. Bohužel se v této verzi nenachází oblíbený nástroj Enterprise Manager, pomocí kterého lze spravovat a monitorovat všechny aplikace a systémy z jediného integrovaného administrátorského nástroje.<sup>11</sup> Novou funkcí, která stojí za zmínění a podporuje ji verze 10g je tzv. Grid computing -

---

<sup>11</sup> ORACLE Czech s.r.o. *Databáze Oracle – edice*. [online]. c2007, [cit. 2007-03-25]. Dostupný z WWW: <<http://www.oracle.com/global/cz/database/edice.html>>.

skupina levných serverů propojených za pomoci software Oracle, tedy rozložení výpočetního výkonu mezi více serverů. Cena Standard Edition pro jedno CPU se pohybuje kolem 114 000 Kč.<sup>12</sup>

## Microsoft

Stejně jako společnost Oracle patří SQL servery od společnosti Microsoft ke špičce mezi SQL servery jak už rozsáhlými skupinami funkcí, rychlostí, spolehlivostí tak profesionálním servisem a dokumentací. Poslední verzi serveru je MS SQL Server 2005, samozřejmostí je výběr z více verzí tohoto produktu, uživatel má možnost si vyzkoušet verzi Express Edition která je zdarma. Standard Edition je možné pořídit za cenu kolem 120 000 Kč.<sup>13</sup>

### 3.7.2 Open Source SQL servery

## MySQL

Tento databázový server oblíbený zejména v nasazení na pro WWW aplikace. Je velice rychlý, má dobrou podporu standardů jazyka SQL, ale v základním módu postrádá některé vlastnosti, například transakce, trigger, uložené procedury a pohledy, taky referenční integrita je podporována pouze pro tabulky typu InnoDB, čímž je v podstatě diskvalifikován z vážnějšího nasazení. MySQL je bezproblémově použitelná v UNIX i Windows operačních systémech. I přes tyto nedostatky patří k nejvíce užívaným SQL serverům pro svou jednoduchost, rychlost a přehlednost. Poslední dostupnou Open Source verzí je MySQL 5.2. Společnost MySQL AB také vyvíjí komerční produkty MySQL Enterprise.<sup>14</sup>

---

<sup>12</sup> STICKFISH s.r.o - 64bit.cz: *Oracle 10g Database Standard Edition One - licence na jedno CPU*. [online]. c2007 [cit. 2007-03-26]. Dostupný z WWW: <<http://www.64bit.cz/oracle-10g-database-standard-edition-one-licence-na-jedno-cpu/209/product.html>>.

<sup>13</sup> MICROSOFT CORPORATION CZ. *Přehled produktu SQL Server 2005*. [online]. c2007, poslední aktualizace 7.11.2005 [cit. 2007-03-26]. Dostupný z WWW: <<http://www.microsoft.com/cze/windowsserversystem/sql/prodinfo/overview/default.mspx>>.

<sup>14</sup> OLŠAVSKÝ, M. *Postgre – Historie a pohledy jinam*. [online]. c2003-2007, poslední aktualizace 12.8.2004 [cit. 2007-03-26]. Dostupný z WWW: <[http://www.linuxsoft.cz/article.php?id\\_article=304](http://www.linuxsoft.cz/article.php?id_article=304)>.

## **PostgreSQL**

Tento server vyniká stabilitou, rychlostí, dobrou podporou a dobrou integrací pokročilých technologií. Server byl primárně určen pro použití v UNIX-ových operačních systémech, donedávna existovala jen alfa verze pro Windows. Ale od verze PostgreSQL 7.5 je i v tomto přístupu změna. Poslední dostupnou verzí tohoto produktu je PostgreSQL 8.2.3. Verze z rady 8.2 podporují již velkou část standardu SQL2003, samozřejmě jako starší verze SQL99 a SQL92.

## **Firebird**

Tento SQL server patří ke skupině poměrně mladých serverových databází i když technologie, kterou nabízí se používá více jak 20 let. Základem pro tento server byla serverová databáze InterBase 6.0. V roce 2000 došlo k uvolnění zdrojových kódů, skupina vývojářů upravila a přepracovala zdrojové kódy a tak vznikl systém Firebird. Poslední stabilní verzí produktu je Firebird 2.0.1 a v současné době je testován systém Firebird 2.1 Alpha. Samozřejmostí pro tyto systémy je podpora uložených procedur a spouští, referenční integrity atd.

Databáze PostgreSQL, MySQL a Firebird nejsou dodávány s žádným grafickým klientem, ke všem existují kvalitní klienty jako externí projekty. Je na výběr jak z klientů napsaných v PHP, tak přímo nativních aplikací. PHPMyAdmin se stejně jako pgAdmin III a FlameRobin používá snadno a intuitivně. Klienty v PHP dokáží s uživatelem komunikovat česky.<sup>15</sup>

---

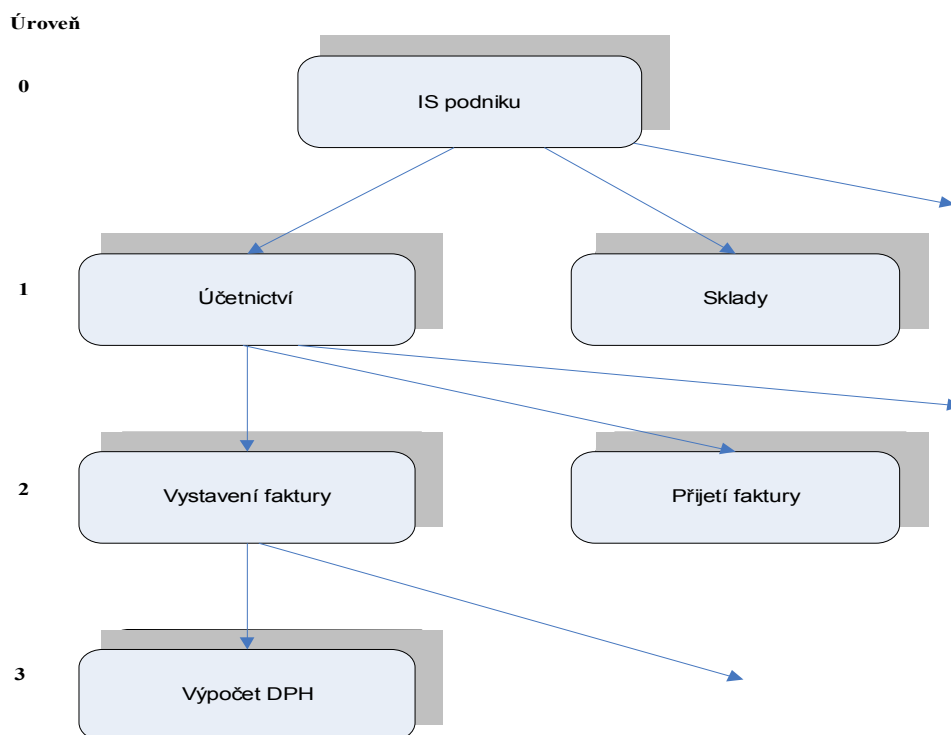
<sup>15</sup> JAKUBČÍK, O. *Srovnání databázových serverů*. [online]. Poslední aktualizace 6.3.2007 [cit. 2007-03-26]. Dostupný z WWW: <<http://www.linuxexpres.cz/software/srovnani-databazovych-serveru>>.



## 4 Vlastní návrhy řešení

V části analýza současného programu pro správu skladového hospodářství byly shrnuty základní nedostatky. V následující části uvedu nástin řešení databázové struktury, ERD diagramy na konceptuální a logické úrovni, závěrem budou shrnuty i možnosti implementačních nástrojů pro vlastní realizaci projektu. V návrhu jsou popsány pouze činnosti a procesy spojené se správou skladu, tedy příjem a výdej ze skladu, přístup pracovníků do skladu, záznam objednávek a tvoření skladových karet pro jednotlivé značky a druhy zboží. Celý model je navržen bez části účetnictví, které se nachází na stejné úrovni v IS, a která úzce souvisí se skladovým hospodářstvím. V hierarchickém modelu informačního systému, bude modelována pouze část Sklady, jak je uvedeno na obrázku níže.

Obrázek č. 1 - Hierarchická struktura IS<sup>16</sup>



<sup>16</sup> KOCH, M. *Datové a funkční modelování*. Brno: Akademické nakladatelství CERM, s.r.o, 2004. s. 65. ISBN 80-214-2724-8.

## **4.1 Seznam požadavků firmy**

- co nejnížší možné náklady,
- přehlednost, jednoduchost návrhu,
- snadné zadávání dat,
- odstranění základních nedostatků.

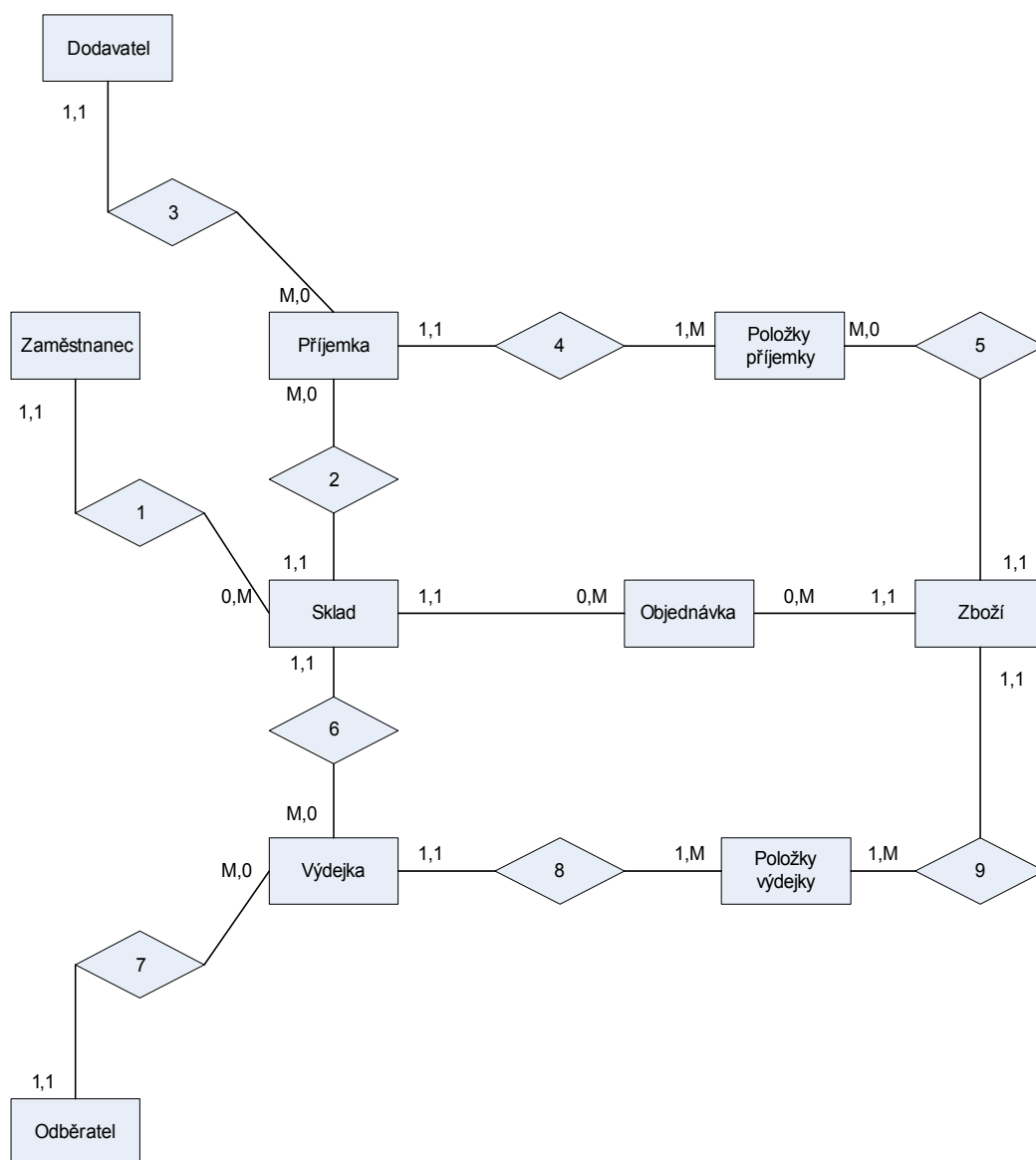
## **4.2 Konceptuální návrh**

Na této úrovni bude podrobně popsána oblast zkoumání pomocí všech hlavních entit, které se v ní vyskytují, a všech vztahů mezi těmito entitami. Entity jsou v datové struktuře reprezentovány tabulkami. Při tomto návrhu není brán v úvahu pozdější způsob implementace ani jakákoliv případná omezení technologického charakteru.

Entity vyskytující se v konceptuálním návrhu jsou pouze hlavní entity, tedy tabulky, které jsou předmětem zájmu dané oblasti informačního systému. V případě návrhu databázové struktury skladu respektive skladového hospodářství je logické, že se jedná o tabulky zboží, příjemka, výdejka, dodavatel atd. Druhou skupinou entit, s kterou bude pracováno a je zahrnuta v relačním schématu logické úrovně jsou číselníky. Číselníky představují tabulky, které slouží jako doplňující údaje pro hlavní datové tabulky a data z těchto tabulek se používají pro kompletnost informací o hlavním datovém objektu (např. dodavatel, zboží). Manipulace s daty v tabulkách reprezentující číselníky není tak častá. Tabulku číselníku je třeba upravit například pouze v případě chceme-li přidat nový druh zboží, který není ještě kategorizován a k určení kategorie nám slouží číselník, který má atributy ID kategorie a název. Doplnění je v tomto případě opravdu jednoduché stačí přidat hodnotu primárního klíče ID kategorie (většinou se jedná o přírůstek čísla předcházejícího, tzv. automatický primární klíč) a název nové kategorie. Typickým příkladem číselníků jsou entity výrobce, jednotka, kategorie, kde u každé položky zboží ve skladu musí být uložena informace od jakého výrobce pochází, v jaké kategorii použití je zařazeno a v jaké měrné jednotce lze zboží počítat. Namísto výpisu výše zmíněných položek v hlavní tabulce, je v hlavní tabulce zboží pouze zahrnut cizí klíč odkazující se na primární klíč příslušného číselníku.

Použití číselníků je velmi elegantní a běžně řešení datové struktury v databázích. Jejich použití má hned několik výhod, jednou z nich je ušetření místa, kdy u každé položky v hlavní tabulce není vypisován celý dlouhý název kategorie například typu varchar délky 30 znaků, ale pouze primární klíč z tabulky číselníku, který je datového typu integer a zabírá pouze 4 bajty. Další bezesporu významnou výhodou je rychlost manipulace s daty, kdy při přejmenování kategorie zboží nemusí uživatel přepisovat název kategorie u každé změněné položky v hlavní tabulce, pouze stačí změnit jednu položku v tabulce představující číselník pro kategorii zboží.

**Obrázek č. 2 - Datový model - konceptuální úroveň**



Tímto konceptuálním návrhem databázové struktury dostaneme obecně platný

popis, který, v případě pozdějšího rozhodnutí zůstane platný, ať už se rozhodneme pro implementaci, která bude vystavěna na jakémkoliv databázovém serveru.

#### **4.2.1 Popis entit a vazeb v ER diagramu**

Název: **Dodavatel / Odběratel**

Každá právnická osoba v případě odběratele i fyzická osoba řádně zaregistrovaná v obchodním rejstříku s kterou je uzavřen obchodně smluvní vztah.

Název: **Příjemka / Výdejka**

Každý doklad, který dokumentuje jednotlivé příjmy nebo výdeje zboží na sklad.

Název: **Zaměstnanec**

Každá fyzická osoba zaměstnaná ve firmě Autolak s pověřením pro správu skladu, výdeje a příjem na sklad.

Název: **Sklad**

Příslušné místo na kterém jsou skladovány zásoby zboží.

Název: **Objednávka**

Záznam požadavku na nákup požadovaného zboží spolu s počtem kusů.

Vazba: **1**

Představuje zaměstnance (#Zaměstnanec) pracujícího ve skladu (#Sklad), s potřebnými pravomocemi na manipulaci se zbožím ve skladu. 1,1 : 0,M.

Vazba: **2 resp. 6**

Sklad (#Sklad) na kterém byla vystavena daná příjemka (#Příjemka), respektive výdejka (#Výdejka). 1,1 : 0,M.

Vazba: **3 resp. 7**

Příjemka (#Příjemka) přijatá od dodavatele zboží (#Dodavatel), respektive výdejka

(#Výdejka) vystavená pro odběratele zboží (#Odběratel). 0,M : 1,1.

Vazba: **4 resp. 8**

Položky (#Položky příjemky), které obsahuje daná příjemka (#Příjemka), respektive položky (#Položky výdejky), které obsahuje daná výdejka (#Výdejka). 1,M : 1,1.

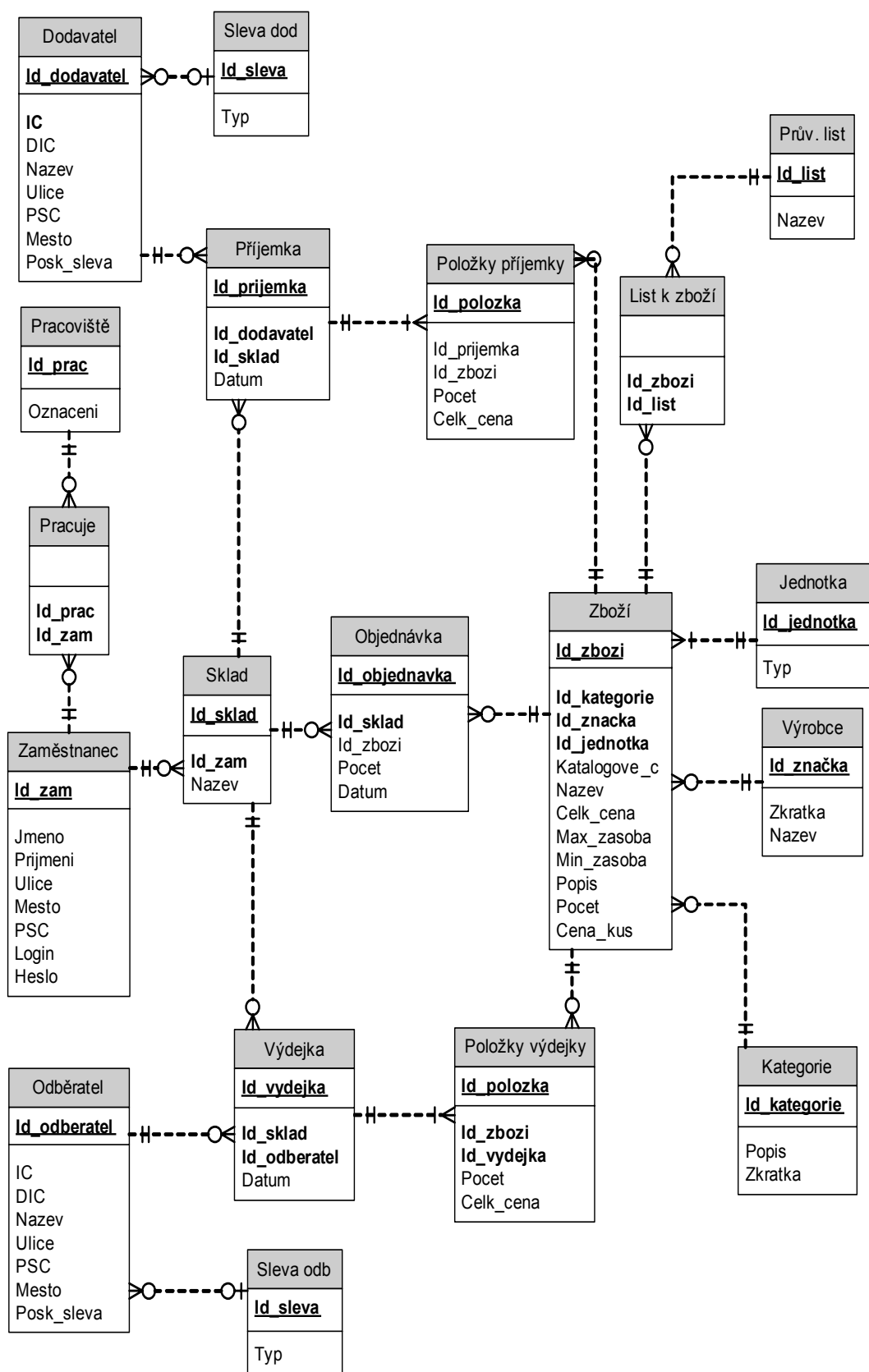
Vazba: **5 resp. 9**

Zboží (#Zboží), jehož příjem na sklad je doložen položkami (#Položky příjemky) příjemky, respektive zboží(#Zboží), jehož výdej ze skladu je doložen položkami výdejky (#Položky výdejky). 1,1 : M,0.

### **4.3 Logické schéma**

Dalším krokem při návrhu jakékoliv databázové struktury po konceptuálním návrhu, je logické relační schéma. Logické schéma je charakteristické vyšší úrovní detailu návrhu. V této úrovni jsou zahrnuty mimo „hlavní“ entity známé již z konceptuální úrovně také tabulky reprezentující číselníky. Logické schéma databáze také obsahuje grafické znázornění relačních vazeb mezi jednotlivými tabulkami.

Obrázek č. 3 - Relační schéma - Logická úroveň



### 4.3.1 Popis atributů entit

**Kategorie** – číselník, tabulka obsahující informace o způsobu využití konkrétního zboží, tedy zda se jedná o krycí materiál, brusivo nebo autolaky.

Id\_kategorie: primární klíč, datový typ Integer.

Zkratka: zkratka pro danou kategorii, datový typ Varchar.

Popis: popis kategorie, datový typ Varchar.

**Výrobce** – číselník, obsahuje všechny značky výrobců daného zboží, které firma Autolak skladuje.

Id\_znacka: primární klíč, datový typ Integer.

Zkratka: zkratka pro výrobce, datový typ Varchar.

Nazev: celý název výrobce, datový typ Varchar.

**Jednotka** – číselník, tabulka obsahuje možné měrné jednotky pro zboží.

Id\_jednotka: primární klíč, datový typ Integer.

Typ: vyjadřuje typ, kterým lze zboží kvantitativně měřit, datový typ Varchar.

**Prův. list** – číselník, tabulka obsahující nezbytné dokumenty (návod na skladování, likvidaci, poučení o hořlavosti).

Id\_list: primární klíč, datový typ Integer.

Nazev: název průvodního listu ke zboží, datový typ Varchar.

**List k zboží** – propojovací tabulka mezi prův. List a Zboží.

Id\_zbozi: cizí klíč z entity Zboží.

Id\_list: cizí klíč z entity Prův. List.

Složený primární klíč – Id\_list, Id\_zbozi.

**Zboží** – hlavní tabulka, obsahuje záznamy zboží, které je skladováno ve firmě s důležitými atributy pro kontrolu zásoby ve skladu i pro ekonomické analýzy.

Id\_zbozi: primární klíč, datový typ Integer.

Id\_kategorie: cizí klíč na entitu Kategorie.

Id\_jednotka: cizí klíč na entitu Jednotka.

Id\_znacka: cizí klíč na entitu Výrobce.

Katalogove\_c.: 10 místné číslo z katalogu dodavatele, datový typ Varchar.

Nazev: název zboží, datový typ Varchar.

Popis: obecný popis zboží, datový typ Varchar.

Pocet: počet kusů ve skladu, datový typ Integer.

Celk\_cena: celková cena všech kusů zboží na skladě bez DPH, datový typ Integer.

Max\_zasoba: maximální udržovaná zásoba daného zboží ve skladu, datový typ Integer.

Min\_zasoba: minimální možná zásoba určitého zboží, která nesmí být překročena, datový typ Integer.

Cena\_kus: cena za jednotku bez DPH, datový typ Integer.

**Položky příjemky** – propojovací tabulka entit Zboží a Příjemka obsahující počet přijímaných kusů zboží a celkovou cenu za tyto kusy bez DPH.

Id\_polozka: primární klíč, datový typ Integer.

Id\_prijemka: cizí klíč na entitu Příjemka.

Id\_zbozi: cizí klíč na entitu Zboží.

Pocet: počet přijímaných kusů zboží, datový typ Integer.

Celk\_cena: celková cena přijímaných kusů zboží bez DPH, datový typ Integer.

**Příjemka** – tabulka, která je nezbytná pro příjem zboží na sklad od dodavatele, obsahuje záznamy interních dokumentů – příjemek.

Id\_prijemka: primární klíč, datový typ Integer.

Id\_sklad: cizí klíč na entitu Sklad.

Id\_dodavatel: cizí klíč na entitu Dodavatel.

Datum: datum vystavení příjemky, datový typ Date.

**Dodavatel** – tabulka shromažďující informace o všech dodavatelích od kterých firma Autolak odebírá zboží.

Id\_dodavatel: primární klíč, datový typ Integer.



IC: Identifikační 8 místné číslo dodavatele, které je zapsáno v obchodním rejstříku, datový typ Varchar.

DIC: DIČ dodavatele, v případě že je vyplněno víme, že dodavatel je plátce DPH, datový typ Varchar.

Nazev: název dodavatele, datový typ Varchar.

Ulice: název ulice na které dodavatel sídlí s číslem popisným, datový typ Varchar.

PSC: poštovní směrovací číslo dodavatele, datový typ Varchar.

Mesto: název města ve kterém dodavatel sídlí, datový typ Varchar.

Posk\_sleva: cizí klíč na entitu Sleva dod.

**Sleva dod.** – číselník, tabulka obsahuje možné typy dodavatelských slev.

Id\_sleva: primární klíč, datový typ Integer.

Typ: typ dodavatelské slevy, která je firmě Autolak poskytována, datový typ Varchar.

**Sklad** – tabulka obsahující informace o skladu, ve kterém je zboží uskladněno.

Id\_sklad: primární klíč, datový typ Integer.

Id\_zam: cizí klíč na entitu Zaměstnanec.

Nazev: jméno skladu ve kterém je zboží uskladněno. Sklad pro velkoprodej, sklad pro přímý maloobchod, sklad pro uskladnění pigmentů barem připravených k míchaní atd.

**Objednávka** – Tabulka se záznamy objednaného množství zboží.

Id\_objednavka: primární klíč, datový typ Integer.

Id\_sklad: cizí klíč na entitu Sklad.

Id\_zbozi: cizí klíč na entitu Zboží.

Pocet: požadované množství k objednání, datový typ Integer.

Datum: datum vznesení požadavku, datový typ Date.

**Zaměstnanec** – tabulka shromažďující informace o všech zaměstnancích firmy Autolak.

Id\_zam: primární klíč, datový typ Integer.

Jmeno: jméno zaměstnance, datový typ Varchar.

Prijmeni: příjmení zaměstnance, datový typ Varchar.

Ulice: název ulice na které zaměstnanec bydlí s číslem popisným, datový typ Varchar.

PSC: poštovní směrovací číslo zaměstnance, datový typ Varchar.

Mesto: název města ve kterém zaměstnanec bydlí, datový typ Varchar.

Login: přístupové jméno zaměstnance do IS, datový typ Varchar.

Heslo: přístupové heslo zaměstnance do IS, datový typ Varchar.

**Pracuje** – propojovací tabulka entitami Zaměstnanec a Pracoviště.

Id\_zam: cizí klíč na entitu Zaměstnanec.

Id\_prac: cizí klíč na entitu Pracoviště.

Složený primární klíč – Id\_zam, Id\_prac.

**Pracoviště** – Pracoviště nebo část skladu na které je zaměstnanec firmy Autolak přiřazen.

Id\_prac: primární klíč, datový typ Integer.

Oznaceni: popis cílového pracoviště, datový typ Varchar.

**Položky výdejky** – propojovací tabulka entit Zboží a Výdejka obsahující počet vydaných kusů zboží a celkovou cenu za tyto kusy bez DPH.

Id\_polozka: primární klíč, datový typ Integer.

Id\_vydejka: cizí klíč na entitu Výdejka.

Id\_zbozi: cizí klíč na entitu Zboží.

Pocet: počet vydaných kusů zboží, datový typ Integer.

Celk\_cena: celková cena vydaných kusů zboží bez DPH, datový typ Integer.

**Výdejka** – tabulka, která je nezbytná pro výdej zboží ze skladu pro koncového odběratele.

Id\_vydejka: primární klíč, datový typ Integer.

Id\_sklad: cizí klíč na entitu Sklad.

Id\_odberatel: cizí klíč na entitu Odběratel.

Datum: datum vystavení výdejky, datový typ Date.

**Odběratel** – tabulka shromažďující informace o všech odběratelích, kterým firma Autolak dodává zboží.

Id\_odberatel: primární klíč, datový typ Integer.

IC: identifikační 8 místné číslo odběratele, které je zapsáno v obchodním rejstříku, datový typ Varchar.

DIC: DIČ odběratele, v případě že je vyplněno víme, že odběratel je plátce DPH, datový typ Varchar.

Nazev: název odběratele, datový typ Varchar.

Ulice: název ulice na které odběratel sídlí s číslem popisným, datový typ Varchar.

PSC: poštovní směrovací číslo odběratele, datový typ Varchar.

Mesto: název města ve kterém odběratel sídlí, datový typ Varchar.

Posk\_sleva: cizí klíč na entitu Sleva odb.

**Sleva odb.** – číselník, tabulka obsahuje možné typy odběratelských slev.

Id\_sleva: primární klíč, datový typ Integer.

Typ: typ odběratelské slevy, kterou firma Autolak poskytuje, datový typ Varchar.

#### **4.4 Návrh tabulek a normální formy**

Většina tabulek, které jsou navrženy v logickém návrhu splňuje 3. Normální Formu (3NF), není nutné podrobovat entity dalším normálním formám, v praxi je 3NF dostačující stupeň normalizace. Tímto by se měla minimalizovat redundance dat, eliminovat nebezpečí vzniku nekonzistence při modifikacích a vyloučení všech netriviálních funkčních závislostí mezi jednotlivými atributy entit.

V návrhu se také vyskytují entity, které nesplňují 3. normální formu, jsou to entity Dodavatel, Odběratel a Zaměstnanec. Při postoupení zmíněných entit do

normálních forem by bylo zapotřebí přesunout atributy - ulice, mesto, PSC do nové tabulky, která by měla vazbu na ony entity. Tyto tabulky jsem nechal bez důkladné normalizace z důvodu, že se jedná o malý podnik, který zaměstnává minimální počet zaměstnanců a jeho odběratelská respektive dodavatelská síť neobsahuje stovky kontaktů. Při důsledné normalizaci bychom museli přidávat například tabulku PSC, která by obsahovala tisíce záznamů PSC celé České republiky, což v našem případě nemá smysl. V některých případech může i striktní dodržování normálních forem způsobit databázovou strukturu nepřehlednou a hlavně zpomalit celý systém.

Diskutabilní jsou také tabulky obsahující atribut Datum, tento atribut by měl být podle teorie také rozložen na jednotlivé údaje jako den, měsíc, rok ale většina databázového softwaru tento atribut uchovává jako jeden údaj a jsou schopny z něj získat kteroukoliv časovou jednotku. U atributu Datum je uveden datový typ Date, protože databázový návrh je stavěn obecně pro implementaci na jakýkoliv databázový server. V prostředí různých databázových serverů může datový typ atributu Datum označen jako Datetime, Timestamp atd.

## **4.5 Návrh softwarového vybavení**

Při hledání produktů jsem se spíše soustředil na Open Source produkty, které se v mnoha ohledech vyrovnají drahým komerčním produktům. Zaměření na Open Source produkty, bylo hlavně z důvodu požadavků firmy Autolak+ spol. s r.o. a to co nejnižších nákladů na vytvoření a poté následnou správu databáze. Vybral jsem databázový server MySQL.

K jeho zvolení z možných kandidátů přispělo několik významných faktorů. Jedním z faktorů je velmi snadná instalace, v dnešní době již existují programové balíčky (např. EasyPHP) které v sobě obsahují tři programy a to konkrétně MySQL databázi, PHP (a PHPMyAdmin) a Apache server. V tomto programovém balíčku tedy dostaneme webový server, databázový server a podporu programovacího jazyku PHP. V případě, že by společnost chtěla dále navrhnout a vytvořit IS nebo WWW stránky například s objednávkovými formuláři, nepotřebuje již jiný software. V našem případě

pro návrh databáze zboží firmy bude stačit MySQL server spolu s grafickým klientem PHPMyAdmin, který podstatně zjednodušuje a zrychluje práci při tvorbě databáze. Další z důvodů výběru MySQL je kompletní podpora českého jazyka. Mezi další nesporné výhody také patří neustálý vývoj a zlepšování databázového serveru MySQL, kdy od verze 5.x došlo k výraznému zlepšení oproti předchozím verzím. Je zde plně funkční integritní omezení pomocí cizích klíčů, transakční zpracování, dále obsahuje uložené procedury, triggerly apod.

Nevýhodou Open Source produktů je absence analytických funkcí. Které jsou většinou v široké míře podporovány u komerčních produktů.

## 5 Přínos (efektivnost) návrhů řešení

Implementace navrhované databázové struktury by měla přinést výrazné zefektivnění a zrychlení procesu prodeje zboží zákazníkovi, kdy zboží v databázi skladu může být vyhledáno nejenom podle katalogového čísla ale také podle kategorie použití a výrobce zboží.

Navržená databáze skladu by měla sloužit jako první krok k návrhu nového informačního systému ve firmě. Kde by měl následovat postupně návrh dalších nezbytných částí IS a vytvořily by se WWW stránky s objednávkovými formuláři a prezentací pro profesionální lakovací autoservisy.

Dalším důležitým přínosem je identifikace pracovníka a zavedení systému práv. Aby pracovník mohl pracovat v navržené databázi je nezbytné jeho přihlášení. Na základě přihlášení můžeme kontrolovat které dokumenty pracovník vystavil. Nová databáze by měla také zajistit lepší optimalizaci skladu, konkrétně snadnější kontrolu maximálních a minimálních zásob na skladu, tak aby nevznikaly ekonomické ztráty příliš velkými zásobami.

Bezesporu k významným ekonomickým přínosům pro firmu, které byly stanoveny také jako požadavky firmy, patří využití nekomerčního softwaru, který je třeba pro navrhované řešení. Použití databázového serveru MySQL přináší úspory pořizovacích nákladů, protože jsou distribuované zdarma i pro komerční účely. Při použití téměř srovnatelných produktů MS SQL Server nebo Oracle Database by se pořizovací náklady vyšplhaly do hladiny kolem 100 000 Kč. Tyto finanční výdaje by pro malý podnik jako je firma Autolak byly zbytečně vysokým zatížením.

## 6 Závěr

V úvodní části bakalářské práce je společnost Autolak+ spol. s.r.o. stručně představena. Nechybí nejdůležitější odběratelské, dodavatelské subjekty a konkurenti společnosti, k úvodu také patří SWOT analýza.

V další úvodní kapitole se zaměřuji na analýzu současného stavu z hlediska práce uživatelů se stávající databází zboží a také na stručné nastínění vybavení společnosti výpočetní technikou. V této kapitole jsou shrnuty základní nedostatky a slabá místa, která byla sestavena na základě pohovorů se zaměstnanci firmy i praktické zkušenosti při práci ve firmě během povinné odborné praxe.

Po úvodní části následuje teoretická část práce. Poznatky z literatury v této části byly kromě obecného seznámení se s pojmem databáze a jazykem SQL, zaměřeny hlavně na normalizaci databáze, která je nezbytným prvkem k správné funkčnosti celého systému, bez redundance dat a celkové konzistentnosti databáze. Závěr této kapitoly je věnován popisu a možnostem výběru správného databázového serveru s rozdělením na komerční a Open Source produkty.

Poslední kapitolou jsou návrhy na zlepšení. V této části se věnuji návrhu databázové struktury skladového hospodářství ve firmě Autolak+ spol. s.r.o. Databázová struktura je tvořena podle požadavků společnosti. Jeden ze specifikovaných cílů je přehlednost, ten je zajištěn vytvořením unikátních automatických primárních klíčů v každé tabulce databáze. I když je v tabulce vhodný kandidát na primární klíč, například v tabulce Zboží - katalogové číslo produktu (atribut katalogove\_c), tak je vytvořen právě automatický (přírůstek čísla předcházejícího) primární klíč, který zajistí snadnou orientaci v celé databázi. Další z požadavků, který jsem řešil v návrhové části byl, aby bylo možné produkty vyhledávat podle různých kategorií použití, výrobců atd.

Při výběru databázového serveru jsem se orientoval na Open Source programy, které mají tzv. otevřený kód a jsou distribuovány zdarma i pro komerční použití, z důvodu požadavku co nejnižších nákladů byl vybrán databázový server MySQL.

## Seznam obrázků a tabulek

Obrázek č. 1 - Hierarchická struktura IS .....	33
Obrázek č. 2 - Datový model - konceptuální úroveň.....	35
Obrázek č. 3 - Relační schéma - Logická úroveň.....	38
Tabulka č. 1 - Nesplnění 1NF .....	26
Tabulka č. 2 - 3NF, kdy opravdu normalizovat?.....	27



## Seznam použité literatury

### *Písemné zdroje publikované*

(1) HERNANDEZ, M., VIESCAS, J. *Myslíme v jazyku SQL : Tvorba dotazů*. 1. vyd. Praha: Grada Publishing, 2004. 380 s. ISBN 80-247-0899-X.

(2) KOCH, M. *Datové a funkční modelování*. 1. vyd. Brno: Akademické nakladatelství CERM, s.r.o, 2004. 108 s. ISBN 80-214-2724-8.

(3) KOSEK, J. *PHP: tvorba interaktivních internetových aplikací*. 1. vyd. Praha: Grada Publishing, 1998. 492 s. ISBN 80-7169-373-1.

(4) LACKO, L. *SQL : Hotová řešení*. 1. vyd. Brno : Computer Press, 2003. 296 s. ISBN 80-7226-975-5.

(5) POWELL, G. *Beginning database design*. 1. vyd. Wrox, 2005. 504 s. ISBN 0-7645-7490-6.

### *Internetové zdroje*

(6) EUROEKONOM. *Svět ekonomie, obchodu, investic a liberalismu v citátech*. [online]. Poslední aktualizace 15.03.2007 [cit. 2007-05-22]. Dostupný z WWW: <<http://www.euroekonom.cz/citaty.html>>.

(7) JAKUBČÍK, O. *Srovnání databázových serverů*. [online]. Poslední aktualizace 6.3.2007 [cit. 2007-03-26]. Dostupný z WWW: <<http://www.linuxexpres.cz/software/srovnani-databazovych-serveru>>.

(8) KOCAN, M. *Víte, co je SQL?* [online]. c2007, poslední aktualizace 26.10. 1998 [cit. 2007-03-20]. Dostupný z WWW: <<http://www.zive.cz/h/Programovani/AR.asp?ARI=3320>>.

- (9) MICROSOFT CORPORATION CZ. *Přehled produktu SQL Server 2005*. [online]. c2007, poslední aktualizace 7.11.2005 [cit. 2007-03-26]. Dostupný z WWW: <<http://www.microsoft.com/cze/windowsserversystem/sql/prodinfo/overview/default.mspx>>.
- (10) OLŠAVSKÝ, M. *Postgre – Historie a pohledy jinam*. [online]. c2003-2007, poslední aktualizace 12.8.2004 [cit. 2007-03-26]. Dostupný z WWW: <[http://www.linuxsoft.cz/article.php?id\\_article=304](http://www.linuxsoft.cz/article.php?id_article=304)>.
- (11) ORACLE Czech s.r.o. *Databáze Oracle – edice*. [online]. c2007, [cit. 2007-03-25]. Dostupný z WWW: <<http://www.oracle.com/global/cz/database/edice.html>>.
- (12) SOCHOR, J. *Analýza a návrh systémů* [online]. Brno: FI MU, 2002. Dostupný z WWW: <<http://www.fi.muni.cz/~sochor/PB007/>>.
- (13) STICKFISH s.r.o - 64bit.cz. *Oracle 10g Database Standard Edition One - licence na jedno CPU*. [online]. c2007 [cit. 2007-03-26]. Dostupný z WWW: <<http://www.64bit.cz/oracle-10g-database-standard-edition-one-licence-na-jedno-cpu/209/product.html>>.
- (14) *Teorie relačních databází : Normalizace*. [online]. c2005-2007, poslední aktualizace 19.5. 2006 [cit. 2007-03-24]. Dostupný z WWW: <<http://www.manualy.net/article.php?articleID=13>>.
- (15) ŽÁK, K. *Historie relačních databází*. [online]. c1998-2007, poslední aktualizace 19.10. 2001 [cit. 2007-03-15]. Dostupný z WWW: <<http://www.root.cz/clanky/historie-relacnich-databazi/>>.

### ***Firemní materiály***

- (16) AUTOLAK+ spol. s.r.o. *Žádost o obnovení úvěrových smluv 2006*. 56 s. 31.3.2006.

## Seznam použitých zkratk

**SWOT** – analýza, která hodnotí silné (Strenghts) a slabé (Weaknesses) stránky, příležitosti (Opportunities) a hrozby (Threats) spojené s určitým projektem, typem podnikání.

**SQL** – Struktured Query Language – strukturovaný dotazovací jazyk používaný pro práci s daty v databázích.

**LAN** – Local Area Network – malá lokální síť např. v rámci jednoho podniku.

**IS** – informační systém.

**SŘBD** – Systém řízení báze dat – softwarové vybavení, které zajišťuje práci s databází, tzn. tvoří rozhraní mezi aplikačními programy a uloženými daty.

**GUI** – Graphical User Interface – grafické uživatelské rozhraní.

**PHP** – Hypertext Preprocesor – skriptovací programovací jazyk, především pro tvorbu dynamických internetových stránek.

**CPU** – Central Procesing Unit – procesor, výkonná jednotka počítače.

**ERD** – Entity Relationship Diagrams – Entito-relační model, je nástroj pro popis entit a jejich vztahů.

# Přílohy

## *Seznam příloh*

Příloha č. 1 – Příkazy SQL pro vytvoření navrhované databáze v MySQL 5.x ....	53
Příloha č. 2 - Vybrané procedury z navrhované databáze skladu .....	58

## **Příloha č. 1 – Příkazy SQL pro vytvoření navrhované databáze v MySQL 5.x**

```
CREATE DATABASE `Autolak_sklad`;  
USE `Autolak_sklad`;
```

```
CREATE TABLE `Zaměstnanec` (  
  `Id_zam` INT( 3 ) NOT NULL AUTO_INCREMENT,  
  `Jmeno` VARCHAR( 15 ) NOT NULL,  
  `Prijmeni` VARCHAR( 20 ) NOT NULL,  
  `Ulice` VARCHAR( 25 ) NOT NULL,  
  `PSC` VARCHAR( 7 ) NOT NULL,  
  `Mesto` VARCHAR( 20 ) NOT NULL,  
  `Login` VARCHAR( 10 ) NOT NULL,  
  `Heslo` VARCHAR( 10 ) NOT NULL,  
  PRIMARY KEY ( `Id_zam` )  
) TYPE = InnoDB;
```

```
CREATE TABLE `Pracoviště` (  
  `Id_prac` INT( 2 ) NOT NULL AUTO_INCREMENT,  
  `Oznaceni` VARCHAR( 30 ) NOT NULL,  
  PRIMARY KEY ( `Id_prac` )  
) TYPE = InnoDB;
```

```
CREATE TABLE `Pracuje` (  
  `Id_prac` INT( 2 ) NOT NULL REFERENCES `Pracoviště` ( `Id_prac` ),  
  `Id_zam` INT( 3 ) NOT NULL REFERENCES `Zaměstnanec` ( `Id_zam` ),  
  PRIMARY KEY ( `Id_zam`, `Id_prac` )  
) TYPE = InnoDB;
```

```
CREATE TABLE `Dodavatel` (  
  `Id_dodavatel` INT(3) NOT NULL AUTO_INCREMENT,  
  `IC` VARCHAR(8) NOT NULL,  
  `DIC` VARCHAR(10),
```

```

`Nazev` VARCHAR(25) NOT NULL,
`Ulice` VARCHAR(25) NOT NULL,
`PSC` VARCHAR(7) NOT NULL,
`Mesto` VARCHAR(20) NOT NULL,
`Posk_sleva` INT(2) NOT NULL REFERENCES `Sleva dod` (`Id_sleva`),
PRIMARY KEY (`Id_dodavatel`)
) TYPE = InnoDB

```

```

CREATE TABLE `Sleva dod` (
`Id_sleva` INT( 2 ) NOT NULL AUTO_INCREMENT,
`Typ` VARCHAR( 15 ) NOT NULL ,
PRIMARY KEY ( `Id_sleva` )
) TYPE = InnoDB;

```

```

CREATE TABLE `Příjemka` (
`Id_prijemka` INT(10) NOT NULL AUTO_INCREMENT,
`Id_sklad` INT(2) NOT NULL REFERENCES `Sklad` (`Id_sklad`),
`Id_dodavatel` INT(3) NOT NULL REFERENCES `Dodavatel` (`Id_dodavatel`),
`Datum` DATE NOT NULL,
PRIMARY KEY (`Id_prijemka`)
) TYPE = InnoDB;

```

```

CREATE TABLE `Položky příjemky` (
`Id_polozka` INT(5) NOT NULL AUTO_INCREMENT,
`Id_zbozi` INT( 10 ) NOT NULL REFERENCES `Zboží` (`Id_zbozi`),
`Id_prijemka` INT(10) NOT NULL REFERENCES `Příjemka` (`Id_prijemka`),
`Pocet` INT(8) NOT NULL,
`Celk_cena` INT(10) NOT NULL,
PRIMARY KEY (`Id_polozka`)
) TYPE = InnoDB

```

```

CREATE TABLE `List k zboží` (

```

```

`Id_list` INT( 2 ) NOT NULL REFERENCES `Prův. list` (`Id_list`),
`Id_zbozi` INT( 10 ) NOT NULL REFERENCES `Zboží` (`Id_zbozi`),
PRIMARY KEY ( `Id_zbozi`, `Id_list` )
) TYPE = InnoDB;

```

```

CREATE TABLE `Prův. list` (
`Id_list` INT( 2 ) NOT NULL AUTO_INCREMENT,
`Nazev` VARCHAR( 15 ) NOT NULL,
PRIMARY KEY ( `Id_list` )
) TYPE = InnoDB;

```

```

CREATE TABLE `Sklad` (
`Id_sklad` INT(2) NOT NULL AUTO_INCREMENT,
`Id_zam` INT( 3 ) NOT NULL REFERENCES `Zaměstnanec` (`Id_zam`),
`Nazev` VARCHAR(20) NOT NULL,
PRIMARY KEY ( `Id_sklad` )
) TYPE = InnoDB;

```

```

CREATE TABLE `Objednávka` (
`Id_objednavka` INT(5) NOT NULL AUTO_INCREMENT,
`Id_sklad` INT(2) NOT NULL REFERENCES `Sklad` (`Id_sklad`),
`Id_zbozi` INT( 10 ) NOT NULL REFERENCES `Zboží` (`Id_zbozi`),
`Pocet` INT(8) NOT NULL,
`Datum` DATE NOT NULL,
PRIMARY KEY ( `Id_objednavka` )
) TYPE = InnoDB

```

```

CREATE TABLE `Zboží` (
`Id_zbozi` INT( 10 ) NOT NULL AUTO_INCREMENT,
`Id_kategorie` INT( 3 ) NOT NULL REFERENCES `Kategorie` (`Id_kategorie`),
`Id_znacka` INT( 3 ) NOT NULL REFERENCES `Výrobce` (`Id_znacka`),
`Id_jednotka` INT( 1 ) NOT NULL REFERENCES `Jednotka` (`Id_jednotka`),

```

```

`Katalogove_c` VARCHAR( 12 ) NOT NULL,
`Nazev` VARCHAR( 25 ) NOT NULL,
`Celk_cena` INT( 6 ) NOT NULL,
`Max_zasoba` INT( 10 ) NOT NULL,
`Min_zasoba` INT( 10 ) NOT NULL,
`Popis` VARCHAR( 40 ) NOT NULL,
`Pocet` INT( 10 ) NOT NULL,
`Cena_kus` INT(6) NOT NULL,
PRIMARY KEY ( `Id_zbozi` )
) TYPE = InnoDB;

```

```

CREATE TABLE `Jednotka` (
`Id_jednotka` INT( 1 ) NOT NULL AUTO_INCREMENT,
`Typ` VARCHAR( 15 ) NOT NULL,
PRIMARY KEY ( `Id_jednotka` )
) TYPE = InnoDB;

```

```

CREATE TABLE `Výrobce` (
`Id_znacka` INT( 3 ) NOT NULL AUTO_INCREMENT,
`Zkratka` VARCHAR( 3 ) NOT NULL,
`Nazev` VARCHAR( 20 ) NOT NULL,
PRIMARY KEY ( `Id_znacka` )
) TYPE = InnoDB;

```

```

CREATE TABLE `Kategorie` (
`Id_kategorie` INT( 3 ) NOT NULL AUTO_INCREMENT,
`Zkratka` VARCHAR( 3 ) NOT NULL,
`Popis` VARCHAR( 20 ) NOT NULL,
PRIMARY KEY ( `Id_kategorie` )
) TYPE = InnoDB;

```

```

CREATE TABLE `Položky výdejky` (

```



```

`Id_polozka` INT(5) NOT NULL AUTO_INCREMENT,
`Id_zbozi` INT( 10 ) NOT NULL REFERENCES `Zboží` (`Id_zbozi`),
`Id_vydejka` INT(10) NOT NULL REFERENCES `Výdejka` (`Id_vydejka`),
`Pocet` INT(8) NOT NULL,
`Celk_cena` INT(10) NOT NULL,
PRIMARY KEY (`Id_polozka`)
) TYPE = InnoDB

```

```

CREATE TABLE `Výdejka` (
`Id_vydejka` INT(10) NOT NULL AUTO_INCREMENT,
`Id_sklad` INT(2) NOT NULL REFERENCES `Sklad` (`Id_sklad`),
`Id_odberatel` INT(3) NOT NULL REFERENCES `Odběratel` (`Id_odberatel`),
`Datum` DATE NOT NULL,
PRIMARY KEY (`Id_vydejka`)
) TYPE = InnoDB;

```

```

CREATE TABLE `Odběratel` (
`Id_odberatel` INT(3) NOT NULL AUTO_INCREMENT,
`IC` VARCHAR(8) NOT NULL,
`DIC` VARCHAR(10) NOT NULL,
`Nazev` VARCHAR(25) NOT NULL,
`Ulice` VARCHAR(25) NOT NULL,
`PSC` VARCHAR(7) NOT NULL,
`Mesto` VARCHAR(20) NOT NULL,
`Posk_sleva` INT(2) NOT NULL REFERENCES `Sleva odb` (`Id_sleva`),
PRIMARY KEY (`Id_odberatel`)
) TYPE = InnoDB

```

```

CREATE TABLE `Sleva odb` (
`Id_sleva` INT( 2 ) NOT NULL AUTO_INCREMENT,
`Typ` VARCHAR( 15 ) NOT NULL ,
PRIMARY KEY ( `Id_sleva` )

```

) TYPE = InnoDB;

## **Příloha č. 2 - Vybrané procedury z navrhované databáze skladu**

### **Vyhledání zboží podle kategorie, značky výrobce a ceny**

CREATE PROCEDURE vyb\_1

BEGIN

DECLARE kat\_zk VARCHAR(3);

DECLARE vyr\_zk VARCHAR(3);

DECLARE cena INT(6);

SELECT z.katalogove\_c, z.cena\_kus, z.pocet

FROM zboží z, výrobce v, kategorie k

WHERE z.id\_kategorie=k.id\_kategorie AND z.id\_znacka=v.id\_znacka

AND z.cena\_kus<=cena AND k.zkratka=kat\_zk AND výrobce.zkratka=vyr\_zk

END;

### **Vyhledání příjemky podle zadaného data a názvu dodavatele**

CREATE PROCEDURE vyb\_2

BEGIN

DECLARE datum DATE;

DECLARE dodavatel\_nazev VARCHAR(15);

SELECT d.nazev, d.IC, p.id\_prijemka, p.datum, s.nazev

FROM dodavatel d, příjemka p, sklad s

WHERE d.id\_dodavatel=p.id\_dodavatel AND s.id\_klad=p.id\_sklad

AND p.datum=datum AND d.nazev=dodavatel\_nazev

END;

### **Vyhledání objednávky (čísla objed., počtu kusů, katalogového čísla zboží s názvem skladu do kterého má být zboží objednáno) podle data objednání.**

```
CREATE PROCEDURE vyb_3
BEGIN

DECLARE datum DATE;

SELECT o.id_objednavka, o.pocet, z.katalogove_c, z.nazev, s.nazev
FROM Objednavka o, Zboží z, Sklad s
WHERE o.id_zbozi=z.id_zbozi AND o.id_sklad=s.id_sklad AND o.datum=datum
END;
```